DOCUMENT RESUME

ED 385 554 TM 023 971

AUTHOR Kim, Sung-Ho
TITLE An Extension of CART's Pruning Algorithm. Program Statistics Research Technical Report No. 91-11.
INSTITUTION Educational Testing Service, Princeton, N.J.
REPORT NO ETS-RR-91-34
PUB DATE 8 Apr 91
NOTE 31p.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Algorithms; *Decision Making; Equations (Mathematics); Prediction; *Regression (Statistics)
IDENTIFIERS *Binary Trees; *CART Computer Program; Graphic Representation; Pruning (Binary Trees)

ABSTRACT

Among the computer-based methods used for the construction of trees such as AID, THAID, CART, and FACT, the only one that uses an algorithm that first grows a tree and then prunes the tree is CART. The pruning component of CART is analogous in spirit to the backward elimination approach in regression analysis. This idea provides a tool in controlling the tree sizes to some extent and thus estimating the prediction error by the tree within a certain range of tree size. In the CART pruning process, Breiman, Friedman, Olshen, and Stone (1984) use a linear combination of the expected loss of the decisions by the tree and the total number of the terminal nodes of the tree. In this paper, CART's pruning is extended by considering a function of all the nodes of the tree in addition to the factors involved in the linear combination. For example, if the cost of observing a variable at each node is considered as the main concern of this paper, or the structural complexity of the tree, such an extension can be seen. (Contains two figures and six references.) (Author)

RR-91-34

An Extension of CART's Pruning Algorithm

Sung-Ho Kim
Educational Testing Service

PROGRAM STATISTICS RESEARCH

TECHNICAL REPORT NO. 91-11

Educational Testing Service
Princeton, New Jersey 08541

An Extension of CART's Pruning Algorithm

Sung-Ho Kim
Educational Testing Service
Princeton, NJ 08541

April 8, 1991

This paper is based on a part of my dissertation. I appreciate Kikumi Tatsuoka and Howard Wainer for their comments on a draft of this paper.

The Program Statistics Research Technical Report Series is designed to make the working papers of the Research Statistics Group at Educational Testing Service generally available. The series consists of reports by the members of the Research Statistics Group as well as their external and visiting statistical consultants.

Abstract

Among the computer-based methods used for the construction of trees such as AID, THAID, CART and FACT, the only one that uses an algorithm that first grows a tree and then prunes the tree is CART. The pruning component of CART is analogous in spirit to the backward elimination approach in regression analysis. This idea provides a tool in controling the tree sizes to some extent and thus estimating the prediction error by the tree within a certain range of tree size. In the CART pruning process, Breiman, Friedman, Olshen, and Stone (1984) use a linear combination of the expected loss of the decisions by the tree and the total number of the terminal nodes of the tree. In this paper, CART's pruning is extended by considering a function of all the nodes of the tree in addition to the factors involved in the linear combination. For example, if we consider the cost of observing a variable at each node as is the main concern of this paper, or the structural complexity of the tree, we can see such an extention.

Key Words: decision-support tree; optimal pruning; the smallest optimally pruned subtree; sufficient tree.

1 Introduction and Motivation

Consider a sequential decision making problem where observations are made sequentially depending upon the outcome of the previous observation, and after each observation a decision is to be made on whether to continue observation or to stop observing and make a final decision about the dependent (or response) variable. If we depict this sequential process from the first observation of a random variable through to the final decisions in a graph, we will end up with a tree-like structure, under the condition that the observations are made on categorical variables only. We will call such a graph a tree. We define a tree in a graphics terminology as a connected, directed and acyclic graph where there is only one path from one vertex to another, and the direction indicates the sequence of observations. The graph (a) of Figure 2.1 in section 2 is an illustration of a tree, where observations are made at the circles and a box symbolizes a final decision. We will call the circles the nodes, and the boxes the terminal nodes.

Trees are among the data analysis tools (factor analysis, nonparametric scaling, and so forth) that have been proposed by social and biomedical scientists motivated by the need to cope with actual data problems involving large numbers of variables. In particular, Breiman, Friedman, Olshen, and Stone (1984) note that the tree-structured methods are very competent in finding a classification rule when the complexity of a data set includes aspects such as high dimensionality, a mixture of

data types (e.g., quantitative and qualitative, or from different stochastic models), variation of dimensions over elements in the data set, or nonhomogeneity.

The use of trees in regression analysis dates back to the Automatic Interaction Detection program (AID) developed at the Institute for Social Research, University of Michigan, by Morgan and Sonquist (1963), which was followed by the classification program THAID, developed by Morgan and Messenger (1973). Breiman et al. (1984) proposed an algorithm, which they called *Classification And Regression Trees*, that is designed as a sequential decision aid for classification or regression problems. Given appropriate data, CART provides a guide, in a form of an upside-down tree, for the order in which to observe predictor variables, when to stop observation, and what decision to make about an interested yet-unknown outcome. The computer program that is based on this algorithm is referred to as CART. Loh and Vanichsetakul (1988) subsequently proposed an algorithm called *Fast Algorithm for Classification Trees* which involves recursive application of linear discriminant analysis, with the predictor variables at each stage being appropriately chosen according to the data and the type of splits desired. The computer program based on the algorithm is called FACT.

The algorithms that underly AID, THAID and FACT grow a tree by adding in branches (variables) as long as a particular condition holds. In contrast, the CART algorithm constructs a tree in two steps — first, growing it and then pruning it. In general terms, CART uses a loss function in the growing process, which ends when

the expected loss no longer decreases (i.e., remains the same). Then, in the CART pruning process, the number of the terminal nodes of a tree is considered in addition to the loss function, and the process ends when a linear combination of the number of terminal nodes and the expected loss of the tree is minimized. A tree constructed in this manner has several desirable properties. Since we use a single loss function as a criterion in the growing process, we can read from the tree which predictor variable is more informative (conditional on some other predictor variables) based on the loss function. The grow-then- prune approach avoids trees being too small or too large compared with the trees constructed by a top-down stopping method (see Section 3.1 of Breiman et al. (1984) and Breiman and Friedman (1988)).

CART's pruning is analogous to the backward elimination in regression analysis. As the latter is proposed as a remedy for stopping too early in the regression model searching process, so is the former as a remedy for stopping with a too small tree. In CART's pruning, we consider the number of the terminal nodes as a complexity penalty of the tree. By specifying the penalty rate, we can control the number of the terminal nodes within a certain range. In other words, the pruning reduces the tree to a certain range of tree sizes.

As mentioned above, CART's pruning deals with the terminal nodes only. A motivation for an extension of CART's pruning is that we may expand our attention from the terminal nodes to all the nodes of a tree in the pruning process. In this paper, we will consider the observation cost of the variables at the nodes along

with the tree size in the pruning process. Here, we can expect an argument against the idea of considering the observation cost at the pruning process rather than the growing process. But, if we include the observation cost at the growing process, the tree will be less informative about the concerned data structure compared with the tree grown using a single loss function. The information would be blurred by adding an extraneous factor to the loss function. In this paper, we will derive some results that are useful in developing an optimal pruning algorithm using a linear combination of the loss function, the number of the terminal nodes, and a function of non-terminal nodes.

The remainder of this paper is organized in 5 sections. In Section 2, we specify the basic notation and definitions concerning trees. In Section 3, we discuss the basic properties of the function used in pruning. Section 4 is the main part of the paper. In it we derive an optimal pruning algorithm under our extended situation. Section 5 gives a summary and a brief comment on a possible application of the idea behind the methods of Section 4 to other pruning criteria.

2 Notation

We borrow most of the notation used here from Breiman et al. (1984). For a tree τ, We let $\tilde{\tau}$ be the set of all the terminal nodes of tree τ and $N(\tau)$ the set of all the non-terminal nodes of tree τ. For a set A, we let $|A|$ be the number of

elements in A. A *subtree* of a tree τ is a tree as a part of τ. For $t \in N(\tau)$, we denote by τ_t the subtree of τ whose root node is t and whose terminal nodes are those of τ that follow from the node t. For notational convenience, when a tree notation has a subscript as in $\tau_{(\cdot)}$, say, we write $\tau_{(\cdot)t}$ for a subtree of $\tau_{(\cdot)}$ whose root node is $t \in N(\tau_{(\cdot)})$. For a non-trivial tree τ and a non-terminal node t of τ, we denote by $\tau \setminus \tau_t$ the pruned subtree of τ which is obtained by cutting τ_t off of τ while leaving the node t on the tree.

{Figure 2.1 about here.}

If we denote by L the loss function which compares a prediction made from the tree and the corresponding outcome of interest, then the conditional expected loss of the prediction for the outcome given the results of the predictor variables, $X_1 = 1$ and $X_2 = 1$, say, is given by

$$E(L|X_1 = 1, X_2 = 1).$$

Without loss of generality, we may assume that the predictor variables are finitely discrete. We denote by r(t) the conditional expected loss at the node t (i.e., when we condition on the event described by the predictors up to that node) and we let

$$R(t) = P(t) \cdot r(t),$$

where P(t) is the arrival rate at node t in the tree τ (i.e., the probability that the tree send a subject from the root node to the node t). Then, the risk in using

predictions from the tree τ is

$$R(\tau) = \sum_{t \in \tilde{\tau}} R(t). \tag{2.1}$$

If t is not the root node of a non-trivial tree τ, then we denote

$$R(\tau_t) = \sum_{s \in \tilde{\tau}_t} R(s), \tag{2.2}$$

where R(s)'s are obtained based on the tree τ.

3 Preliminaries

As we indicated at the end of Section 1, we wish to consider the cost or the time of the variable observation at each non-terminal node, and we denote this cost at the node t by W_t (W for *weight*). We assume $W_t > 0$ for a non-terminal node t and, $W_t = 0$ when t is a terminal node since no observation is made there. We define a cost function for the pruning process which involves W_t and investigate it in this section.

We denote by par(t) the node which immediately precedes the node t. For a node t of a tree τ which is not the root node, we let

$$W(t) = \sum W_s,$$

where the summation goes over the set of all the nodes on the path from the root node through par(t). When t is the root node, we let $W(t) = 0$. Then, we have the following result.

Theorem 3.1 *For any tree* τ,

$$(a) \qquad \sum_{t \in \tilde{\tau}} P(t)W(t) = \sum_{t \in N(\tau)} P(t)W_t.$$

For a node $t \in \tau$,

$$(b) \qquad \sum_{s \in \tilde{\tau}_t} P(s)W(s) = \sum_{s \in N(\tau_t)} P(s)W_{s+s'(t)}W(t).$$

Proof: The proof of (a) is by induction. If τ is trivial, we have

$$W(t) = 0 = W_t, \quad \text{for } t \in \tilde{\tau} = \{t\}.$$

Suppose that the result holds for $\tau = \tau'$ and that τ' is branched at a terminal node t' into a new tree τ'' such that $\tilde{\tau}'' = \tilde{\tau}' \cup \{t'_1, t'_2, \cdots, t'_a\}$. Then,

$$\begin{aligned} \sum_{t \in \tilde{\tau}''} P(t)W(t) &= \sum_{t \in \tilde{\tau}'} P(t)W(t) - P(t')W(t') + \sum_{s \in \{t'_1, t'_2, \cdots, t'_a\}} P(s)W(s) \\ &= \sum_{t \in N(\tau')} P(t)W_t - P(t')W(t') + \sum_{s \in \{t'_1, t'_2, \cdots, t'_a\}} P(s)W(s), \end{aligned}$$

where the second equality follows from the supposition. The last term on the right hand side of the last equation is equal to

$$P(t')W(t'_1) = P(t')(W(t') + W_{t'}).$$

Thus, we have

$$\begin{aligned} \sum_{t \in \tilde{\tau}''} P(t)W(t) &= \sum_{t \in N(\tau')} P(t)W_t + P(t')W_{t'} \\ &= \sum_{t \in N(\tau'')} P(t)W_t. \end{aligned}$$

The proof of part (b) is direct:

$$\begin{aligned}\sum_{s\in\tilde{\tau}_t} P(s)W(s) &= \sum_{s\in\tilde{\tau}_t} P(s)(W(s)-W(t)+W(t)) \\ &= \sum_{s\in\tilde{\tau}_t} P(s)(W(s)-W(t))+P(t)W(t) \\ &= \sum_{s\in N(\tau_t)} P(s)W_s + P(t)W(t),\end{aligned}$$

where the last equality follows from (a). □

For non-negative real numbers β_i, $i = 1, 2$, we now define a new risk function for our pruning process:

$$R_{\underset{\sim}{\beta}}(\tau) = R(\tau) + \beta_1 \sum_{t\in\tilde{\tau}} P(t)W(t) + \beta_2|\tilde{\tau}|, \tag{3.1}$$

where $\underset{\sim}{\beta} = (\beta_1, \beta_2)$. Thus $R_{\underset{\sim}{\beta}}(\tau)$ is a linear combination of the risk function $R(\tau)$ used in growing the tree, the number of terminal nodes of the tree $|\tilde{\tau}|$, and the expected value of $W(t)$ for the terminal nodes t. For any node $t \in N(\tau) \cup \tilde{\tau}$, let

$$R_{\underset{\sim}{\beta}}(t) = R(t) + \beta_1 P(t)W(t) + \beta_2. \tag{3.2}$$

and

$$R_{\underset{\sim}{\beta}}(\tau_t) = R(\tau_t) + \beta_1 \sum_{s\in\tilde{\tau}_t} P(s)W(s) + \beta_2|\tilde{\tau}_t|. \tag{3.3}$$

The following theorem is straightforward.

Theorem 3.2 *Let τ be a non-trivial tree. Then, for $t \in N(\tau)$,*

(a) $$R_{\underset{\sim}{\beta}}(\tau) - R_{\underset{\sim}{\beta}}(\tau\backslash\tau_t) = R_{\underset{\sim}{\beta}}(\tau_t) - R_{\underset{\sim}{\beta}}(t).$$

(b) For every ancestor s of the node t,

$$R_{\underset{\sim}{\beta}}(\tau_s) - R_{\underset{\sim}{\beta}}(\tau_s \setminus \tau_t) = R_{\underset{\sim}{\beta}}(\tau_t) - R_{\underset{\sim}{\beta}}(t).$$

Proof: From expressions (2.1) and (2.2), we see that

$$R(\tau \setminus \tau_t) = R(\tau) - R(\tau_t) + R(t). \tag{3.4}$$

By definition, we have

$$|\widetilde{\tau \setminus \tau_t}| = |\tilde{\tau}| - |\tilde{\tau}_t| + 1, \tag{3.5}$$

and

$$\begin{aligned} & R_{\underset{\sim}{\beta}}(\tau \setminus \tau_t) + R_{\underset{\sim}{\beta}}(\tau_t) - R_{\underset{\sim}{\beta}}(t) \\ &= [R(\tau \setminus \tau_t) + R(\tau_t) - R(t)] + \beta_1[\sum_{s \in \widetilde{\tau \setminus \tau_t}} P(s)W(s) + \sum_{s \in \tilde{\tau}_t} P(s)W(s) - P(t)W(t)] \\ & \quad + \beta_2[|\widetilde{\tau \setminus \tau_t}| + |\tilde{\tau}_t| - 1]. \end{aligned} \tag{3.6}$$

But for the expression in the second bracket in expression (3.6), we have

$$\begin{aligned} & \sum_{s \in \widetilde{\tau \setminus \tau_t}} P(s)W(s) + \sum_{s \in \tilde{\tau}_t} P(s)W(s) - P(t)W(t) \\ &= \sum_{s \in N(\tau \setminus \tau_t)} P(s)W_s + \sum_{s \in N(\tau_t)} P(s)W_s + P(t)W(t) - P(t)W(t) \\ &= \sum_{s \in N(\tau)} P(s)W_s \\ &= \sum_{s \in \tilde{\tau}} P(s)W(s), \end{aligned}$$

where the first and the last equalities follow from Theorem 3.1. Hence, by combined expressions (3.1), (3.4) and (3.5), we complete the proof of (a).

The proof of (b) proceeds in the same manner. First we note that

$$R_{\underset{\sim}{\beta}}(\tau_s \setminus \tau_t) + R_{\underset{\sim}{\beta}}(\tau_t) - R_{\underset{\sim}{\beta}}(t)$$

is equal to the right-hand side of expression (3.6) with τ replaced by τ_s. On the other hand,

$$\begin{aligned}
&\sum_{s \in \widetilde{\tau_s \setminus \tau_t}} P(s)W(s) + \sum_{s \in \tilde{\tau}_t} P(s)W(s) - P(t)W(t) \\
&= \sum_{u \in N(\tau_s \setminus \tau_t)} P(u)W_u + P(s)W(s) + \sum_{u \in N(\tau_t)} P(u)W_u + P(t)W(t) - P(t)W(t) \\
&= \sum_{u \in N(\tau_s)} P(u)W_u + P(s)W(s) \\
&= \sum_{u \in \tilde{\tau}_s} P(u)W(u).
\end{aligned}$$

Therefore, the result follows from expressions (3.4) and (3.5) with τ replaced by τ_s.

□

The extended loss function of expression (3.1) is rather difficult to handle, and thus we will develop an alternative version of it below and examine its properties for use in finding an optimal pruning method in Section 4. Let

$$\underset{\sim}{W}(\tau_t) = \sum_{s \in \tilde{\tau}_t} P(s)W(s) - P(t)W(t). \tag{3.7}$$

For a fixed non-trivial tree τ and a node $t \in N(\tau)$, we let

$$g_1(t, \tau) = \frac{R(t) - R(\tau_t)}{\underset{\sim}{W}(\tau_t)} \tag{3.8}$$

and

$$g_2(t, \tau) = \frac{R(t) - R(\tau_t)}{|\tilde{\tau}_t| - 1}. \tag{3.9}$$

In equations (3.8) and (3.9), we note that, for a non-trivial tree τ,

$$|\tilde{\tau}_t| - 1 \geq 1 \quad \text{and} \quad \underline{W}(\tau_t) > 0. \tag{3.10}$$

Since a branching gives rise to at least two new nodes, the first inequality of expression (3.10) is obvious. Rewriting equation (3.7) gives

$$\underline{W}(\tau_t) = \sum_{s \in \tilde{\tau}_t} P(s)(W(s) - W(t)),$$

where $W(s) - W(t) > 0$ for $s \neq t$ since each individual weight is positive. On the other hand, we never grow a tree when the R-value does not decrease. In other words, a branching is made at a terminal node t of a current tree, i.e., a simple tree τ', whose root node is t and whose terminal nodes are the child nodes of t, is attached to t, only when

$$R(t) > R(\tau'). \tag{3.11}$$

Therefore, we have that, for every non-terminal node t of τ, both $g_1(t,\tau)$ and $g_2(t,\tau)$ are positive.

We now let

$$\Delta_{\underset{\sim}{\beta}}(g_1(t,\tau), g_2(t,\tau)) = g_2(t,\tau) - \beta_2 - \beta_1 \frac{g_2(t,\tau)}{g_1(t,\tau)}, \tag{3.12}$$

and we focus on the difference in risk

$$D_{\underset{\sim}{\beta}}(t,\tau) = R_{\underset{\sim}{\beta}}(t) - R_{\underset{\sim}{\beta}}(\tau_t).$$

Then, we have

$$\Delta_{\underset{\sim}{\beta}}(g_1(t,\tau), g_2(t,\tau)) = D_{\underset{\sim}{\beta}}(t,\tau)/(|\tilde{\tau}_t| - 1). \tag{3.13}$$

From equations (3.10) and (3.13), we can see that, for any non-terminal node t of a non-trivial tree τ,

$$\text{sign}[\Delta_{\underset{\sim}{\beta}}(g_1(t,\tau), g_2(t,\tau))] = \text{sign}[D_{\underset{\sim}{\beta}}(t,\tau)]. \tag{3.14}$$

Because of equation (3.14), we may use the Δ-function of equation (3.12) in place of $R_{\underset{\sim}{\beta}}$ to find an optimal pruning method since the increase or decrease of $R_{\underset{\sim}{\beta}}$, as given by the sign of $D_{\underset{\sim}{\beta}}(t,\tau)$, determines where to prune and when to stop the pruning process. 5

4 Extended Optimal Pruning

We denote by $\tau' \preceq \tau$ the relationship that τ' is a pruned subtree of τ, and by $\tau' \prec \tau$ that τ' is a strictly pruned subtree of τ, i.e., when τ' is a pruned subtree of τ and $\tau' \neq \tau$. We call τ_1 an *optimally pruned subtree* (OPST) of a non-trivial tree τ with respect to $\underset{\sim}{\beta}$ if

$$R_{\underset{\sim}{\beta}}(\tau_1) = \min_{\tau' \preceq \tau} R_{\underset{\sim}{\beta}}(\tau'),$$

and we denote by $\tau(\underset{\sim}{\beta})$ the smallest OPST of τ with respect to $\underset{\sim}{\beta}$.

The following theorem is immediate from the transitivity of the relationship $\preceq$.

Theorem 4.1 *If* $\tau(\underset{\sim}{\beta}) \preceq \tau' \preceq \tau$, *then* $\tau(\underset{\sim}{\beta}) = \tau'(\underset{\sim}{\beta})$.

For notational convenience, we write $\Delta_{\underset{\sim}{\beta}}(t,\tau)$ for $\Delta_{\underset{\sim}{\beta}}(g_1(t,\tau), g_2(t,\tau))$. The

following theorem provides us a convenient algebraic tool to deal with $R_{\underset{\sim}{\beta}}(\cdot)$.

Theorem 4.2 *Let $\tau' \preceq \tau$, where τ' is not a trivial tree. Suppose t is a non-terminal node of τ'. Let $t_i \in \tilde{\tau}'_t \cap N(\tau)$, for $i = 1, 2, \cdots, r$, where $r = |\tilde{\tau}'_t \cap N(\tau)|$. Then*

$$\Delta_{\underset{\sim}{\beta}}(t, \tau') = \Delta_{\underset{\sim}{\beta}}(t, \tau) + \sum_{i=1}^{r} (\Delta_{\underset{\sim}{\beta}}(t, \tau) - \Delta_{\underset{\sim}{\beta}}(t_i, \tau)) \frac{|\tilde{\tau}_{t_i}| - 1}{|\tilde{\tau}'_t| - 1}. \tag{4.1}$$

Proof:

$$\begin{aligned}
& R_{\underset{\sim}{\beta}}(t) - R_{\underset{\sim}{\beta}}(\tau'_t) \\
&= R(t) - R(\tau'_t) - (\beta_1 \underline{W}(\tau'_t) + \beta_2(|\tilde{\tau}'_t| - 1)) \\
&= R(t) - (R(\tau_t) - \sum_{i=1}^{r} (R(\tau_{t_i}) - R(t_i))) - \beta_1 (\underline{W}(\tau_t) - \sum_{i=1}^{r} \underline{W}(\tau_{t_i})) \\
&\quad - \beta_2 (|\tilde{\tau}_t| - \sum_{i=1}^{r} (|\tilde{\tau}_{t_i}| - 1) - 1) \\
&= R(t) - R(\tau_t) - \beta_1 \underline{W}(\tau_t) - \beta_2 (|\tilde{\tau}_t| - 1) - \sum_{i=1}^{r} (R(t_i) - R(\tau_{t_i}) \\
&\quad - \beta_1 \underline{W}(\tau_{t_i}) - \beta_2 (|\tilde{\tau}_{t_i}| - 1)),
\end{aligned}$$

where the first equality follows from expressions (3.2), (3.3) and (3.7). Then, from equation (3.13), we have

$$\begin{aligned}
& R_{\underset{\sim}{\beta}}(t) - R_{\underset{\sim}{\beta}}(\tau'_t) \\
&= \Delta_{\underset{\sim}{\beta}}(t, \tau)(|\tilde{\tau}_t| - 1) - \sum_{i=1}^{r} \Delta_{\underset{\sim}{\beta}}(t_i, \tau)(|\tilde{\tau}_{t_i}| - 1).
\end{aligned} \tag{4.2}$$

Dividing both sides of (4.2) by $(|\tilde{\tau}'_t| - 1)$ gives

$$\Delta_{\underset{\sim}{\beta}}(t, \tau') = \Delta_{\underset{\sim}{\beta}}(t, \tau) \frac{|\tilde{\tau}_t| - 1}{|\tilde{\tau}'_t| - 1} - \sum_{i=1}^{r} \frac{|\tilde{\tau}_{t_i}| - 1}{|\tilde{\tau}'_t| - 1}.$$

Since

$$|\tilde{\tau}_t| = |\tilde{\tau}'_t| + \sum_{i=1}^{r}(|\tilde{\tau}_{t_i}| - 1),$$

the desired result follows. □

Recall that $\beta_1 = 0$ in the CART method. The following corollary is immediate from Theorem 4.2.

Corollary 4.1 *Under the set-up of Theorem 4.2, if* $\beta_1 = 0$, *then*

$$g_2(t,\tau') = g_2(t,\tau) + \sum_{i=1}^{r}(g_2(t,\tau) - g_2(t_i,\tau))\frac{|\widetilde{\tau_{t_i}}| - 1}{|\widetilde{\tau'_t}| - 1}. \tag{4.3}$$

From equation (4.3), we can determine the exact value of $g_2(t,\tau') - g_2(t,\tau)$, rather than whether the inequality $g_2(t,\tau') - g_2(t,\tau) > 0$ is satisfied (see Theorem 10.11 of Breiman et al. (1984)).

Given $\underset{\sim}{\beta}$ and a non-trivial tree τ, we can get a sequence of pruned subtrees of τ with the corresponding $\tau(\underset{\sim}{\beta})$. Let $\tau_{(0)} = \tau$, and

$$\mu_0(\underset{\sim}{\beta}) = \min_{t \in N(\tau_{(0)})}\{\Delta_{\underset{\sim}{\beta}}(t,\tau_{(0)})\}.$$

We define a sequence of trees $\tau_{(i)}$ and the corresponding numbers $\mu_i(\underset{\sim}{\beta})$, for $i = 1,2,\cdots,w$ (w a finite number) sequentially as follows:

Definition 4.1 *Let* τ_i *be such that*

$$N(\tau_{(i)}) = N(\tau_{(i-1)}) - \{N(\tau_{(i-1)t});\ \Delta_{\underset{\sim}{\beta}}(t,\tau_{(i-1)}) = \mu_{i-1}(\underset{\sim}{\beta})\} \tag{4.4}$$

for $i = 1, 2, \cdots, w$, *and then let*

$$\mu_i(\underset{\sim}{\beta}) = \min_{t \in N(\tau_{(i)})} \{\Delta_{\underset{\sim}{\beta}}(t, \tau_{(i)})\}, \quad \textit{for } i = 1, 2, \cdots, w. \tag{4.5}$$

We can continue to apply equations (4.4) and (4.5) sequentially until we reach the trivial tree (i.e., root(τ)). Let $\tau_{(w)} = root(\tau)$.

The following theorem is a big step towards the aim of establishing an algorithm of an extended version of CART.

Theorem 4.3 *For a non-trivial tree* τ, *suppose* $\tau_{(0)}, \tau_{(1)}, \cdots, \tau_{(w)} = root(\tau)$ *are obtained as in Definition 4.1. Then*

(a) $\tau_{(0)} \succ \tau_{(1)} \succ \cdots \succ \tau_{(w)}$.

(b) For $t \in N(\tau_{(i)})$, $i = 1, 2, \cdots, w-1$,

$$\begin{aligned} \Delta_{\underset{\sim}{\beta}}(t, \tau_{(i-1)}) &= \Delta_{\underset{\sim}{\beta}}(t, \tau_{(i)}) \quad \textit{if } \tau_{(i)t} = \tau_{(i-1)t}, \\ \Delta_{\underset{\sim}{\beta}}(t, \tau_{(i-1)}) &< \Delta_{\underset{\sim}{\beta}}(t, \tau_{(i)}) \quad \textit{if } \tau_{(i)t} \prec \tau_{(i-1)t}. \end{aligned} \tag{4.6}$$

Proof: Part (a) follows directly from Definition 4.1. Let $t_1, t_2, \cdots, t_r \in \tilde{\tau}_{(i)} \cap N(\tau_{(i-1)})$. For the node t in (b), we can think of two cases. They are (1) $\tau_{(i)t} = \tau_{(i-1)t}$, and (2) $\tau_{(i)t} \prec \tau_{(i-1)t}$. In case (1), the result is immediate. In case (2), by Definition 4.1, we have

$$\Delta_{\underset{\sim}{\beta}}(t, \tau_{(i-1)}) - \Delta_{\underset{\sim}{\beta}}(t_j, \tau_{(i-1)}) > 0, \quad \text{for } j = 1, 2, \cdots, r.$$

Therefore, part (b) follows from Theorem 4.2. □

In particular, if $\beta_1 = 0$, then we can say, by Corollary 4.1, under the condition of Theorem 4.3, that,

$$\begin{aligned} g_2(t, \tau_{(i-1)}) &= g_2(t, \tau_{(i)}) \text{ if } \tau_{(i)t} = \tau_{(i-1)t}, \\ g_2(t, \tau_{(i-1)}) &< g_2(t, \tau_{(i)}) \text{ if } \tau_{(i)t} \prec \tau_{(i-1)t}. \end{aligned} \tag{4.7}$$

The result (4.7) is well harnessed in the CART method. However, when $\beta_1 \neq 0$, the result (4.7) is of no use.

The following result, which is useful in finding the smallest OPST $\tau(\underset{\sim}{\beta})$, is immediate from Theorem 4.3.

Corollary 4.2 *Let τ be a non-trivial tree. Suppose we obtain $\{\tau_{(i)}\}$ be obtained as in Definition 4.1 for some $\underset{\sim}{\beta}$. Then*

$$\mu_{i-1}(\underset{\sim}{\beta}) < \mu_i(\underset{\sim}{\beta}), \text{ for } i = 1, 2, \cdots, w.$$

Proof:

$$\begin{aligned} \mu_i(\underset{\sim}{\beta}) &= \min_{t \in N(\tau_{(i)})} \{\Delta_{\underset{\sim}{\beta}}(t, \tau_{(i)})\} \\ &> \min_{t \in N(\tau_{(i)})} \{\Delta_{\underset{\sim}{\beta}}(t, \tau_{(i-1)})\} \quad \text{(by Theorem 4.3)} \\ &\geq \min_{t \in N(\tau_{(i-1)})} \{\Delta_{\underset{\sim}{\beta}}(t, \tau_{(i-1)})\} \\ &= \mu_{i-1}(\underset{\sim}{\beta}). \ \square \end{aligned}$$

The following result proves that the set of the pruned subtrees obtained as in

Definition 4.1 contains the smallest OPST.

Theorem 4.4 *Let τ be a non-trivial tree. Suppose that we obtain $\{\tau_{(i)}\}$ as in Definition 4.1 for some $\underset{\sim}{\beta}$. Then*

$$\tau(\underset{\sim}{\beta}) \in \{\tau_{(0)}, \tau_{(1)}, \cdots, \tau_{(w)}\}.$$

Proof: For any τ' such that

$$\tau(\underset{\sim}{\beta}) \prec \tau' \preceq \tau_{(0)}, \tag{4.8}$$

we have, by Theorem 4.1, that

$$\tau(\underset{\sim}{\beta}) = \tau'(\underset{\sim}{\beta}).$$

For any τ' satisfying expression (4.8),

$$\min_{t \in N(\tau')} \{\Delta_{\underset{\sim}{\beta}}(t, \tau')\} \leq 0.$$

Otherwise, there must exist a subtree τ' satisfying expression (4.8) such that $R_{\underset{\sim}{\beta}}(\tau') < R_{\underset{\sim}{\beta}}(\tau'(\underset{\sim}{\beta})) = R_{\underset{\sim}{\beta}}(\tau(\underset{\sim}{\beta}))$, which is a contradiction.

By the definition of $\tau(\underset{\sim}{\beta})$, we have that for every $t \in N(\tau(\underset{\sim}{\beta}))$,

$$\Delta_{\underset{\sim}{\beta}}(t, \tau(\underset{\sim}{\beta})) > 0.$$

Therefore, we can find the smallest OPST $\tau(\underset{\sim}{\beta})$ in the set of $\tau_{(0)}, \tau_{(1)}, \cdots, \tau_{(w)}$, which is obtained in the process of Definition 4.1. □

Theorem 4.4 implies that we have only to look at $\tau_{(0)}, \tau_{(1)}, \cdots, \tau_{(w)}$ to find $\tau(\underset{\sim}{\beta})$. But $\{\mu_i(\underset{\sim}{\beta})\}_{i=0}^{w}$ is more useful for finding $\tau(\underset{\sim}{\beta})$ through the monotonicity of $\mu_i(\underset{\sim}{\beta})$ as shown in Corollary 4.2.

Theorem 4.5 *Let τ be a non-trivial tree. Suppose $\{\tau_{(i)}\}$ are obtained as in Definition 4.1 for some $\underset{\sim}{\beta}$. If, for some i^*, $1 \le i^* \le w$,*

$$\mu_{i^*-1}(\underset{\sim}{\beta}) \le 0 \text{ and } \mu_{i^*}(\underset{\sim}{\beta}) > 0,$$

then

$$\tau(\underset{\sim}{\beta}) = \tau_{(i^*)}.$$

Proof: Suppose, for $i = 1, 2, \cdots, w$, there exist $t_1, t_2, \cdots, t_{r_i} \in \tilde{\tau}_{(i)} \cap N(\tau_{(i-1)})$. Then by repeated use of Theorem 3.2 (a), we have

$$\begin{aligned} R_{\underset{\sim}{\beta}}(\tau_{(i)}) - R_{\underset{\sim}{\beta}}(\tau_{(i-1)}) &= \sum_{j=1}^{r_i} (R_{\underset{\sim}{\beta}}(t_j) - R_{\underset{\sim}{\beta}}(\tau_{(i-1)t_j})) \\ &= \sum_{j=1}^{r_i} \Delta_{\underset{\sim}{\beta}}(t_j, \tau_{(i-1)}) \cdot (|\tilde{\tau}_{(i-1)t_j}| - 1) \\ &= \mu_{i-1}(\underset{\sim}{\beta}) \sum_{j=1}^{r_i} (|\tilde{\tau}_{(i-1)t_j}| - 1), \end{aligned}$$

where the last equation is apparent by Definition 4.1.

By Corollary 4.2, we have

$$\mu_i(\underset{\sim}{\beta}) < 0 \quad \text{for} \quad i \le i^* - 2, \text{ and}$$

$$\mu_i(\underset{\sim}{\beta}) > 0 \quad \text{for} \quad i \ge i^*.$$

Thus

$$R_{\underset{\sim}{\beta}}(\tau_{(i+1)}) - R_{\underset{\sim}{\beta}}(\tau_{(i)}) \begin{cases} < 0 & \text{for } i \leq i^* - 2 \\ > 0 & \text{for } i \geq i^*. \end{cases}$$

Furthermore

$$R_{\underset{\sim}{\beta}}(\tau_{(i^*)}) - R_{\underset{\sim}{\beta}}(\tau_{(i^*-1)}) \begin{cases} = 0 & \text{if } \mu_{i^*-1}(\underset{\sim}{\beta}) = 0 \\ < 0 & \text{if } \mu_{i^*-1}(\underset{\sim}{\beta}) < 0. \end{cases}$$

Therefore,

$$\min_{0 \leq i \leq w} \{R_{\underset{\sim}{\beta}}(\tau_{(i)})\} = R_{\underset{\sim}{\beta}}(\tau_{(i^*)}),$$

and the result follows from Theorem 4.4. □

We now have the following summarizing result.

Theorem 4.6 *Let τ be a non-trivial tree. Suppose $\{\tau_{(i)}\}$ are obtained as in Definition 4.1 for some $\underset{\sim}{\beta}$. Then*

$$\tau(\underset{\sim}{\beta}) = \begin{cases} \tau_{(k+1)} & \text{if } \mu_k(\underset{\sim}{\beta}) = 0 \\ \tau_{(k)} & \text{if } \mu_k(\underset{\sim}{\beta}) > 0 \text{ and } \mu_{k-1}(\underset{\sim}{\beta}) < 0, \text{ for } k \geq 1 \\ \tau_{(0)} & \text{if } \mu_0(\underset{\sim}{\beta}) > 0. \end{cases}$$

Proof: If $\mu_0(\underset{\sim}{\beta}) > 0$, then the result is obvious. If $\mu_k(\underset{\sim}{\beta}) = 0$, then by Corollary 4.2, it is immediate that

$$\mu_i(\underset{\sim}{\beta}) \leq 0 \text{ for } i \leq k$$

and

$$\mu_i(\underset{\sim}{\beta}) > 0 \text{ for } i \geq k+1.$$

Thus, by Theorem 4.5, $\tau(\underset{\sim}{\beta}) = \tau_{(k+1)}$. By the same theorem, we can see that $\tau(\underset{\sim}{\beta}) = \tau_{(k)}$, when $\mu_k(\underset{\sim}{\beta}) > 0$ and $\mu_{k-1}(\underset{\sim}{\beta}) < 0$, for $k \geq 1$. □

Theorem 4.6 is the main result of this section and the paper. For a given $\underset{\sim}{\beta}$, however, it is by no means desirable to grow a tree far beyond the optimal tree $\tau(\underset{\sim}{\beta})$ before pruning up until $\tau(\underset{\sim}{\beta})$ is reached. Theorem 4.6 is available whenever the tree τ thereof contains $\tau(\underset{\sim}{\beta})$. At the end of Section 10.2, Breiman et al. (1984) discuss a method by which one can find a tree which may not be fully grown involving all the possible predictor variables and which contains $\tau(\underset{\sim}{\beta})$. The following results up to Theorem 4.9 are straightforward extensions of Theorem 10.31, Theorem 10.32, and the subsequent paragraphs in Section 10.2 of Breiman et al. (1984) and thus their proofs will be omitted.

Let $node(s,t)$ denote the set of nodes on the path from node s through node t and $l(s,t)$ the number of connections on the same path, i.e., $l(s,t) = node(s,t) - 1$. For $t \in N(\tau) \cup \tilde{\tau}$, denote by $anc(t,\tau)$ the set of all the ancestors of node t in the tree τ. Define, for a non-terminal node t of a non-trivial tree τ,

$$V_{\underset{\sim}{\beta}}(t) = \min_{s \in anc(t) \cup \{t\}} \{R(s) - \beta_1 W(s,t) - \beta_2(l(s,t)+1)\}.$$

where

$$W(s,t) = \sum_{u \in node(s,t)} P(u) W_u.$$

Theorem 4.7 *Let τ be a non-trivial tree. Then,*

$$V_{\underset{\sim}{\beta}}(t) > 0, \quad \forall \ t \in N(\tau(\underset{\sim}{\beta})).$$

If we define

$$\tau_{suff}(\underset{\sim}{\beta}) = \{t \in N(\tau) \cup \tilde{\tau};\ V_{\underset{\sim}{\beta}}(s) > 0,\ \forall\, s \in anc(t)\}, \tag{4.9}$$

then we have the following result.

Theorem 4.8 *For a given $\underset{\sim}{\beta}$,*

$$\tau(\underset{\sim}{\beta}) \preceq \tau_{suff}(\underset{\sim}{\beta}) \preceq \tau.$$

The tree $\tau_{suff}(\underset{\sim}{\beta})$ contains $\tau(\underset{\sim}{\beta})$, so we don't need to go beyond $\tau_{suff}(\underset{\sim}{\beta})$ before starting pruning toward $\tau(\underset{\sim}{\beta})$.

$V_{\underset{\sim}{\beta}}(\cdot)$ can be defined recursively as in the theorem below.

Theorem 4.9 *For any non-terminal node t of a tree τ and a non-negative vector $\underset{\sim}{\beta}$,*

$$V_{\underset{\sim}{\beta}}(t) = \min\{R(t), V_{\underset{\sim}{\beta}}(par(t))\} - \beta_1 P(t) W_t - \beta_2.$$

5 Concluding Remarks

In CART, Breiman et al. (1984) prune trees using as criterion a linear combination of the risk of the predictions and the total number of the terminal nodes. Here, we

have extended the CART pruning algorithm, in the sense that we consider the cost of the variable observation in addition to the factors used in the CART's pruning criterion, and we have derived results useful for a pruning algorithm under this new criterion. CART's pruning algorithm can thus be viewed as a special case ($\beta_1 = 0$ in expression (3.1)) of the algorithm considered in this paper.

Equation (4.1) of Theorem 4.2 plays a key role in deriving the pruning algorithm. It is a useful algebraic tool in dealing with functions defined on trees. Versions of this equation would be possible under various pruning criteria. For example, Arbab and Miche (1985) considered degree of linearity of a tree as a measure of desirability for trees. Their degree of linearity is represented in terms of the non-linearity measure whicl. is defined as follows:

> Let τ be a tree which is composed of the root node and m major subtrees, $\tau_1, \tau_2, \cdots, \tau_m$ as in Fig. 5.1. Then the non-linearity of the tree τ is given by
>
> $$NL(\tau) = \frac{1}{m} \times \sum_{i=1}^{m} \{NL(\tau_i) + (m - i) \times |N(\tau_i)|\},$$
>
> where $NL(\tau) = 0$ if τ is trivial, and $N(\tau_i)$ are sorted in increasing order of $|N(\tau_i)|$.

Given a tree, the non-linearity of the tree is defined over the set of the major subtrees of the tree. And thus at each non-terminal node t, say, of the tree we can assign a non-linearity measure of the subtree whose root node is t. If we considered

this linearity of a tree in addition to the cost function considered in this paper, our pruning method should be more complicated than the present one. This further extended version of the pruning method seems to be an interesting problem to pursue.

{Figure 5.1 about here.}

References

1. Arbab, B. and Michie, D. (1985) "Generating rules from examples," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1, (ed: A. Joshi), Morgan Kaufmann Publishers Inc., 631-633.

2. Breiman, L. and Friedman, J. H. (1988) Comment on "Tree-structured classification via generalized discriminant analysis," by Loh, Wei-Yin and Vanichsetakul, Nunta, *Journal of American Statistical Association,*, 83, 725-727.

3. Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) *Classification and Regression Trees,* Wadsworth International Group, Belmont, California.

4. Loh, W.-Y. and Vanichsetakul, N. (1988) "Tree-structured classification via generalized discriminant analysis," *Journal of American Statistical Association,* 83, 715-728.

5. Morgan, J. N. and Messenger, R. C. (1973) *THAID: a sequential search program for the analysis of nominal scale dependent variables*, Ann Arbor: Institute for Social Research, University of Michigan.

6. Morgan, J. N. and Sonquist, J. A. (1963) "Problems in the analysis of survey data, and a proposal," *Journal of American Statistical Association*, 58, 415-434.

Figure 2.1

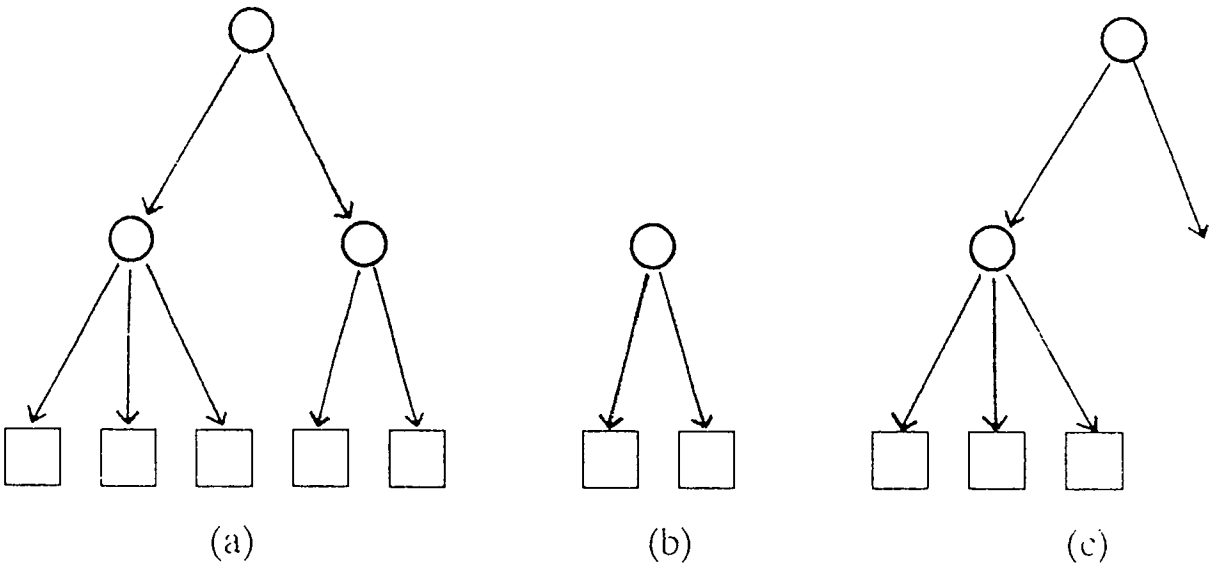

Figure 5.1